The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. A computer for processing information to produce desired results, comprising:

a computer-readable medium for storing services, the services including expressions that represent names by literalizing processes and processes by deliteralizing names; and

a microprocessor for executing services as processes, the processes interacting via names and evolving by performing actions or causing other processes to evolve.

2. The computer of Claim 1, wherein the services include ports identifiable by uniform resource indicators and are undued with behavioral types.

3. The computer of Claim 2, wherein the services compose if a behavioral type of a port of one service is compatible with another behavioral type of another port of another service.

4. The computer of Claim 3, further comprising a decentralized operating system for orchestrating the services for executing on the microprocessor.

5. The computer of Claim 4, further comprising middleware that includes expressions that represent names by literalizing processes and processes by literalizing names, the middleware being interposed between the services and the decentralized operating system.

6. A microprocessor for executing instructions, comprising:

a timing and control unit for retrieving an instruction from memory, decoding the instruction, fetching data connected with the instruction, and saving the result, the data including names obtained by literalizing processes in a reflective process algebra; and

an arithmetic and logic unit for performing an operation specified by the instruction, the instruction being expressed in a reflective process algebra, the reflective process algebra being capable of representing names as literalization of processes and processes as deliteralization of names.

7. The microprocessor of Claim 6, further comprising a register array for storing the result of the executed instruction.

8. The microprocessor of Claim 7, further comprising an instruction register and decoder for holding the instruction of the microprocessor is executing.

9. The microprocessor of Claim 8, further comprising bus connections for allowing the microprocessor to receive data into memory internally and for communicating result of the executed instruction externally.

10. The microprocessor of Claim 9, wherein the timing and control unit, the arithmetic and logic unit, and the instruction register and decoder communicates via ports that have unilateral contracts associated with ports.

11. A computer-readable medium having computer-executable instructions stored thereon for performing a method of allowing processes to act or interact via shared names so as to evolve or cause other processes to evolve, comprising:

computer-executable instructions expressed in a reflective process algebra, the computer-executable instructions including process syntactical elements that represent processes, the process syntactical elements being selected from a group consisting of an inactive process 0, an output process $X[Y]$, an input process $X(Z)\ .\ P$, a lifting process $\text{LIFT } X\ .\ P$, a composition of processes $P|P$, and a deliteralization of a name $\rangle X \langle$.

12. The computer-readable medium of Claim 11, wherein the computer-executable instructions including a syntactical element that represents a name, the syntactical element that represents a name being formed from a literalization of a process $\langle P \rangle$.

13. The computer-readable medium of Claim 12, wherein the name Y in the output process $X[Y]$ is constructed from a literalization of a process P, the literalization of the process P being inhibited if the process P is not a member of a predetermined type T.

14. The computer-readable medium of Claim 12, wherein the name Z in the input process $X(Z) . P$ is constructed from a literalization of a process P, the literalization of the process P being inhibited if the process P is not a member of a predetermined type U.

15. The computer-readable medium of Claim 12, wherein a name X is constructed from the literalization of a process P, the literalization of the process P being inhibited if the size of P exceeds a predetermined threshold.

16. A computer-executable method for processing services comprising:

representing services as processes in a reflective process algebra, the reflective process algebra being capable of expressing processes as deliteralization of names and names as literalization of processes; and

causing processes to act or to communicate when names are shared among processes so as to facilitate the evolution of processes.

17. The computer-executable method of Claim 16, further comprising extracting an expression written in the reflective process algebra from a service and checking to see whether the expression complies with the syntactical rules of the reflective process algebra.

18. The computer-executable method of Claim 17, further comprising determining that the expression is a process if the expression represents a stop process 0.

19. The computer-executable method of Claim 17, further comprising determining that the expression is a process if the expression represents an output process $X[Y]$.

20. The computer-executable method of Claim 17, further comprising determining that the expression is a process if the expression represents an input process $X(Z) . P$.

21. The computer-executable method of Claim 17, further comprising determining that the expression is a process if the expression represents a lift process $\text{LIFT}\ X\ .\ P$.

22. The computer-executable method of Claim 17, further comprising determining that the expression is a process if the expression represents a composition of processes $P|P$.

23. The computer-executable method of Claim 17, further comprising determining that the expression is a process if the expression represents a deliteralization of a name $\rangle X \langle$.

24. The computer-executable method of Claim 17, further comprising determining that the expression is a name if the expression represents a literalization of a process $\langle P \rangle$.

25. The computer-executable method of Claim 17, further comprising determining that the expression is a name if the expression represents a literalization of a process and the size of the process does not exceed a threshold $\langle P \rangle, \#(P) \leq K$.

26. The computer-executable method of Claim 17, further comprising determining that the expression is a name if the expression represents a literalization of a process and the process belongs to a type T as in the syntactical element $\langle P \rangle, P{:}T$.

27. The computer-executable method of Claim 16, further comprising extracting an expression written in the reflective process algebra from a service and checking the expression against a set of equational laws of the reflective process algebra for structural congruence.

28. The computer-executable method of Claim 27, further comprising determining that the expression is structurally congruent to a name X if the expression is of the form $\langle\ \rangle X \langle\ \rangle$.

29. The computer-executable method of Claim 27, further comprising determining that the expression is structurally congruent to a process P if the expression is a composition of processes P|0.

30. The computer-executable method of Claim 27, further comprising determining that the expression is structurally congruent to a process P if the expression is a composition of processes 0|P.

31. The computer-executable method of Claim 27, further comprising determining that the expression is structurally congruent to a composition of processes $P_1 \mid P_0$ if the expression is a composition of processes $P_0 \mid P_1$.

32. The computer-executable method of Claim 27, further comprising determining that the expression is structurally congruent to a composition of processes $P_0 \mid (P_1 \mid P_2)$ if the expression is a composition of processes $(P_0 \mid P_1) \mid P_2$.

33. The computer-executable method of Claim 16, further comprising extracting an expression written in the reflective process algebra from a service and checking the meanings of the expression against a set of operational semantics rules of the reflective process algebra.

34. The computer-executable method of Claim 33, further comprising determining that the expression is reducible to $X\left[\langle P\rangle\right]$ if the expression is of the form LIFT X . P.

35. The computer-executable method of Claim 33, further comprising determining that the expression is reducible to $P_1|P_2$ if the expression is a composition of processes $P_0|P_2$ and P_0 can be reducible to a process P_1.

36. The computer-executable method of Claim 33, further comprising determining that the expression is reducible to a process P_3 if the expression is a process P_2

and P_2 is structurally congruent to P_0 and P_0 can be reducible to P_1 and P_1 is structurally congruent to P_3.

37. The computer-executable method of Claim 33, further comprising determining that the expression is reducible to a second expression $P_0\{X_2/X_1\}$ if the expression is a composition of processes $X_0[X_2]\ |X_0(X_1).P_0$.

38. The computer-executable method of Claim 37, further comprising equating the second expression to an inactive process 0 if the second expression is of a form $(0)\rangle\{\langle Q\rangle/\langle P\rangle\}\langle$.

39. The computer-executable method of Claim 37, further comprising equating the second expression to $(R)\rangle\{\langle Q\rangle/\langle P\rangle\}\ \langle|(S)\rangle\{\langle Q\rangle/\langle P\rangle\}\langle$ if the second expression is of the form $(R|S)\rangle\{\langle Q\rangle/\langle P\rangle\}\langle$.

40. The computer-executable method of Claim 37, further comprising equating the second expression to $(X)\{\langle Q\rangle/\langle P\rangle\}\ (Y)\ .\ ((R)\rangle\{\langle Q\rangle/\langle P\rangle\}\langle\ )$ if the second expression is of the form $(X(Y).R)\rangle\{\langle Q\rangle/\langle P\rangle\}\langle$.

41. The computer-executable method of Claim 37, further comprising equating the second expression to $(X)\{\langle Q\rangle/\langle P\rangle\}\ [(Y)\{\langle Q\rangle/\langle P\rangle\}]$ if the second expression is of the form $(X[Y])\rangle\{\langle Q\rangle/\langle P\rangle\}\langle$.

42. The computer-executable method of Claim 37, further comprising equating the second expression to $\mathrm{LIFT}(X)\{\langle Q\rangle/\langle P\rangle\}.(R)\rangle\{\langle Q\rangle/\langle P\rangle\}\langle$ if the second expression is of the form $(\mathrm{LIFT}\ X.R)\rangle\{\langle Q\rangle/\langle R\rangle\}\langle$.

43. The computer-executable method of Claim 37, further comprising equating the second expression to a process Q if the second expression is of the form $(\rangle X\langle)\rangle\{\langle Q\rangle/\langle P\rangle\}\langle$ and X is structurally congruent to $\langle P\rangle$ and otherwise the act of equating the second expression to $\rangle X\langle$.

44. The computer-executable method of Claim 37, further comprising equating the second expression to a name $\langle Q \rangle$ if the second expression is of the form $(\mathrm{X})\{\langle \mathrm{Q} \rangle / \langle \mathrm{P} \rangle\}$ and X is structurally congruent to $\langle \mathrm{P} \rangle$ and otherwise the act of equating the second expression to X.

45. An array of microprocessors for executing instructions, comprising:

at least one microprocessor that includes one or more components:

a timing and control unit for retrieving an instruction from memory, decoding the instruction, fetching data connected with the instruction, and saving the result, the data including names obtained by literalizing processes in the reflective process algebra; and

an arithmetic and logic unit for performing an operation specified by the instruction, the instruction being expressed in a reflective process algebra, the reflective process algebra being capable of representing names as literalization of processes and processes as deliteralization of names.

46. The array of microprocessors of Claim 45, wherein the array of microprocessors are on a single integrated circuit.

47. The array of microprocessors of Claim 45, wherein the array of microprocessors are on multiple integrated circuits, the multiple integrated circuits being mounted on a single board.

48. The array of microprocessors of Claim 45, wherein the array of microprocessors are on multiple integrated circuits, the multiple integrated circuits being mounted on multiple boards, the multiple boards being housed on a single rack of a computer.

49. The array of microprocessors of Claim 45, wherein the array of microprocessors are on multiple integrated circuits, the multiple integrated circuits being mounted on multiple boards, the multiple boards being housed on multiple racks of multiple computers.

50. The array of microprocessors of Claim 45, further comprising a network for coupling one or more microprocessors, the network being selected from a group consisting of permanent connections and temporary connections.

51. The array of microprocessors of Claim 45, wherein the components of the at least one microprocessor are synchronized by instructions produced by a compiler that compiles a program written in the reflective process algebra.

52. The array of microprocessors of Claim 45, wherein the array of microprocessors are synchronized by instructions produced by a compiler that compiles a program written in the reflective process algebra.

53. The array of microprocessors of Claim 45, wherein the components of the at least one microprocessor lacks circuitry for predicting a next instruction to be executed.